

Matrizes

Introdução à Computação

Professor: Paulo Meirelles (paulormm@ime.usp.br)

Departamento de Ciência da Computação
Instituto de Matemática e Estatística

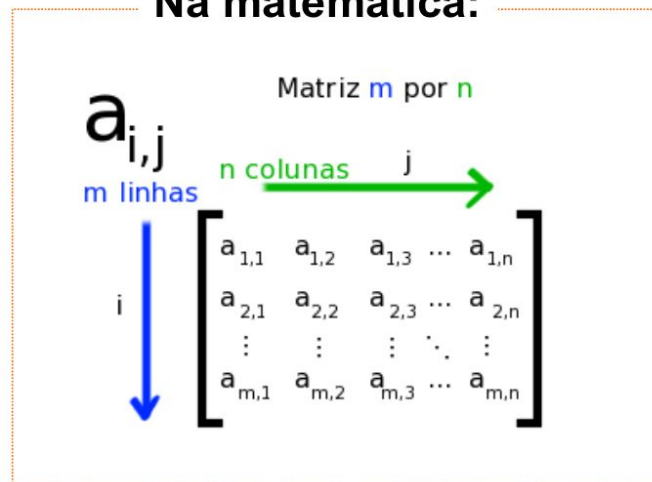


1. Matrices

A Visão Matemática

Em muitas aplicações, precisamos processar dados bidimensionais (gráfico cartesiano de coordenadas, imagens digitais, matrizes de sistemas de equações lineares em Álgebra Linear). Matriz é uma tabela organizada em linhas e colunas no formato $m \times n$, onde **m** representa o número de linhas e **n** o número de colunas.

Na matemática:



No Python

Em Python, matrizes são definidas como listas de listas. Cada linha da matriz em Python é uma lista com n elementos. Por exemplo, o código abaixo cria uma matriz com três linhas e quatro colunas:

```
>>> A = [ [4, 1, 8, 3],  
           [2, 5, 7, 0],  
           [6, 9, 0, 3] ]
```

Índices

Na matemática, $a_{i,j}$ é o elemento que está na linha i e coluna j , com os índices iniciando no 1.

Em Python, os índices começam no zero. Logo, para acessar o elemento da segunda linha e terceira coluna, usamos $i = 1$ e $j = 2$. Sintaxe: `nome_matriz[i][j]`

Exemplos:

```
>>> A[0][0]
```

```
4
```

```
>>> A[0][1]
```

```
1
```

```
>>> A[0][2]
```

```
8
```

```
>>> A[0][3]
```

```
3
```

```
>>> A[1][0]
```

```
2
```

```
>>> A[2][1]
```

```
9
```

Acessando Linhas Completas

A i-ésima linha da matriz pode ser obtida como uma lista usando a sintaxe:

nome_matriz[i]

Exemplos:

```
>>> A[0]  
[4, 1, 8, 3]  
>>> A[1]  
[2, 5, 7, 0]  
>>> A[2]  
[6, 9, 0, 3]
```

Dimensões da Matriz

O número de linhas e colunas de uma matriz podem ser obtidos pela função `len()`.

```
>>> len(A)    #número de linhas
```

```
3
```

```
>>> len(A[0]) #número de colunas
```

```
4
```

Matrizes de Dimensões Quaisquer

O modo explícito de criação de matrizes é inadequado para a criação de matrizes maiores ou com dimensões conhecidas apenas em tempo de execução.

Para criar uma matriz de dimensões arbitrárias, poderíamos pensar em usar o seguinte código:

```
>>> m = 3      #número de linhas
>>> n = 5      #número de colunas
>>> valor = 0   #valor inicial de preenchimento
>>> linha = [valor]*n
>>> matriz = [linha]*m
>>> print(matriz)
[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]
```


As 3 linhas são a mesma linha!?

Ao modificar `matriz[0][0]` na posição $i = 0$ e $j = 0$ da matriz obtemos o seguinte resultado:

```
>>> matriz[0][0] = 1
>>> print(matriz)
[[1, 0, 0, 0, 0], [1, 0, 0, 0, 0], [1, 0, 0, 0, 0]]
```

Estranhamente, os `matriz[1][0]` e `matriz[2][0]` também tiveram seus valores alterados.

O comando `matriz = [linha]*m` gerou uma matriz em que todas linhas são referências para a mesma lista na memória. Ou seja:

```
>>> id(matriz[0]) == id(matriz[1])
True
>>> id(matriz[0]) == id(matriz[2])
True
```

Criação de Matrizes

```
def cria_matriz(m, n, valor):  
    matriz = []  
    linha = [valor]*n  
    for i in range(m):  
        matriz.append(linha[:])  
    return matriz
```

Clonagem de mesma linha

```
def cria_matriz(m, n, valor):  
    matriz = []  
    for i in range(m):  
        linha = []  
        for j in range(n):  
            linha.append(valor)  
        matriz.append(linha)  
    return matriz
```

Construção linha a linha

2. Manipulação de Matrizes

Soma de Matrizes

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

+

$$B = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

$$C = \begin{bmatrix} 11 & 22 & 33 \\ 44 & 55 & 66 \\ 77 & 88 & 99 \end{bmatrix}$$

Inicializa Triângulo de Pascal em Python

No triângulo de Pascal, um elemento é a soma de seus dois superiores:

$$a[i][j] = a[i-1][j-1] + a[i-1][j]$$

Para isso funcionar, a primeira coluna da matriz deve ser inicializada com 1's:

```
for i in range(n+1):
```

$$a[i][0] = 1$$

MAC

Introdução à Computação

- ▶ Professor: Paulo Meirelles
E-mail: paulormm@ime.usp.br